Copy 2 of 2

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE 05/01/2002	3. REPORT TYPE AND DATES COVERED FINAL REPORT 6/1/98-5/31/01

4. TITLE AND SUBTITLE
Engineering Robust Distributed Database Software

5. FUNDING NUMBERS
DAAG55-98-1-0331

6. AUTHOR(S)
PI: Val Tannen, co-PIs: Peter Buneman and Susan Davidson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Pennsylvania
200 S 33rd Street Philadelphia PA 19104

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
38743.1-CI

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The research has focused on a technology for ``mobile information management''. Underlying this technology is a mathematical foundation enabling the use of formal methods in developing and reasoning about the construction of mobile information management components and their use in database integration and transformation. The salient features of our approach are: use of unmaterialized views; dynamic integration of data consumers and data sources, using mobile query processes; data interface specifications, based on XML schemas; specifications for transformations, based on XML query languages; the development of formal methods, focusing on query and constraint reformulation.

14. SUBJECT TERMS
Databases, Distributed Information Management, XML

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

MASTER COPY: PLEASE KEEP THIS "MEMORANDUM OF TRANSMITTAL" BLANK FOR REPRODUCTION PURPOSES. WHEN REPORTS ARE GENERATED UNDER THE ARO SPONSORSHIP, FORWARD A COMPLETED COPY OF THIS FORM WITH EACH REPORT SHIPMENT TO THE ARO. THIS WILL ASSURE PROPER IDENTIFICATION. NOT TO BE USED FOR INTERIM PROGRESS REPORTS; SEE PAGE 2 FOR INTERIM PROGRESS REPORT INSTRUCTIONS.

MEMORANDUM OF TRANSMITTAL

U.S. Army Research Office
ATTN: AMSRL-RO-BI (TR)
P.O. Box 12211
Research Triangle Park, NC 27709-2211

☐ Reprint (Orig + 2 copies)
☐ Technical Report (Orig + 2 copies)
☐ Manuscript (1 copy)
☑ Final Progress Report (Orig + 2 copies)
☐ Related Materials, Abstracts, Theses (1 copy)

CONTRACT/GRANT NUMBER: DAAG55-98-1-0331

REPORT TITLE: Engineering Robust Distributed Database Software

is forwarded for your information.

SUBMITTED FOR PUBLICATION TO (applicable only if report is manuscript): 9+2 pages

Sincerely,

Val Tannen

(VAL TANNEN)
PI

ARO Contract DAAG55-98-1-0331

Engineering Robust Distributed Database Software

06/01/1998-05/31/2001

PI: V. Tannen, co-PIs: P. Buneman, S. Davidson

Final Report

1 Personnel

In addition to the PIs, this contract has partially supported PhD students Alin Deutsch, Wenfei Fan, Carmem Hara, Hartmut Liefke, Lucian Popa, Arnaud Sahuguet, and Wang-Chiew Tan.

2 Goals

The research has focused on a technology for "mobile information management". Underlying this technology is a mathematical foundation enabling the use of formal methods in developing and reasoning about the construction of mobile information management components and their use in database integration and transformation. The salient features of our approach are:

1. Use of unmaterialized views.
2. Dynamic integration of data consumers and data sources, using mobile query processes.
3. Data interface specifications, based on XML schemas.
4. Specifications for transformations, based on XML query languages.
5. The development of formal methods, focusing on query and constraint reformulation.

3 Project Summaries

3.1 Mobile queries and distributed query languages

Sahuguet and Tannen have worked on **ubQL** a new distributed query language for programming large-scale distributed query systems such as resource sharing systems. The language is obtained by adding a small set of mobile process primitives (communication channels, migration operators, etc.) on top of any traditional query language. Queries are encapsulated into processes and can migrate between sites thus enabling cooperation. An important methodological device is the separation of the installation (including migration) of query processes from the distributed execution of the queries. ubQL allows the encoding of widely used distributed query patterns such as chaining, referral, subscription, leasing, recruiting, query/data/hybrid shipping, etc., and evaluate some

language-based rewrite strategies for the installation of ubQL queries that use only partial and distributed knowledge of execution costs.

Sahuguet and Tannen (with Pierce) have worked on new mechanisms in distributed query optimization. This work outlines a flexible framework for optimizing and deploying distributed queries in wide area networks. The database field has developed very powerful techniques for finding efficient execution plans for declaratively specified queries. However, applying these optimization techniques in the setting of distributed information management requires centralized knowledge of the entire network and assumes passive behavior from the data sources. The reality of the Web is different. Future distributed query optimizers must handle (in fact, exploit!) a rich variety of information flow mechanisms like chaining, referral, proxying, brokering, publish-subscribe, leasing, etc. We look to mobile agent technologies for the combination of flexibility and precision needed for handling these mechanisms. Our language-based approach uses a mobile process calculus based on the pi-calculus in combination with a powerful query-plan language. The salient characteristic of the language is that messaging, migration, and database operations all live in the same semantic space and interact, creating new opportunities for optimization.

3.2 Query reformulation and optimization

Popa and Tannen have studied a class of path-conjunctive queries and constraints (dependencies) defined over complex values with dictionaries. This class includes the relational conjunctive queries and embedded dependencies, as well as many interesting examples of complex value and oodb queries and integrity constraints. We show that some important classical results on containment, dependency implication, and chasing extend and generalize to this class.

Deutsch, Popa and Tannen have continued the work on an optimization method and algorithm designed for several objectives: physical data independence, using materialized views/cached queries, semantic optimization, and generalized tableau minimization. The method relies on generalized forms of chase and "backchase" with constraints (dependencies). By using dictionaries (finite functions) in physical schemas we can capture with constraints useful access structures such as indexes, materialized views, source capabilities, access support relations, gmaps, etc. In this reporting period, we have shown that the method is usable in realistic optimizers by extending it to bag and mixed (i.e. bag-set) semantics as well as to grouping views and by showing how to integrate it with standard cost- based optimization. We understand materialized views broadly, including user-defined views, cached queries and physical access structures (such as join indexes, access support relations, and gmaps). Moreover, our internal query representation supports object features hence the method applies to OQL and (extended) SQL:1999 queries. Chase and backchase supports a very general class of integrity constraints, thus being able to find execution plans using views that do not fall in the scope of other methods. In fact, we prove completeness theorems that show that our method will find the best plan in the presence of common and practically important classes of constraints and views, even when bag and set semantics are mixed.

The search space for query plans is defined and enumerated in a novel manner: the chase phase rewrites the original query into a "universal" plan that integrates all the access structures and alternative pathways that are allowed by applicable constraints. Then, the backchase phase produces optimal plans by eliminating various combinations of redundancies, again according to constraints.

This method is applicable (sound) to a large class of queries, physical access structures, and semantic constraints. We prove that it is in fact complete for path-conjunctive queries and views with complex

objects, classes and dictionaries, going beyond previous theoretical work on processing queries using materialized views.

Popa, Deutsch, Sahuguet and Tannen have studied the practicality this novel method for generating alternative query plans that uses chasing (and back-chasing) with logical constraints. The method brings together use of indexes, use of materialized views, semantic optimization and join elimination (minimization). Each of these techniques is known separately to be beneficial to query optimization. The novelty of our approach is in allowing these techniques to interact systematically, eg. non-trivial use of indexes and materialized views may be enabled only by semantic constraints.

We have implemented our method for a variety of schemas and queries. We examine how far we can push the method in term of complexity of both schemas and queries. We propose a technique for reducing the size of the search space by "stratifying" the sets of constraints used in the (back)chase. The experimental results demonstrate that our method is practical (i.e., feasible and worthwhile).

Hara and Davidson have studied functional dependencies for nested data. Functional dependencies add semantics to a database schema, and are useful for studying various problems, such as database design, query optimization and how dependencies are carried into a view. In the context of a nested relational model, these dependencies can be extended by using path expressions instead of attribute names, resulting in a class of dependencies that we call nested functional dependencies (NFDs). NFDs define a natural class of dependencies in complex data structures; in particular they allow the specification of many useful intra- and inter-set dependencies (i.e., dependencies that are local to a set and dependencies that require consistency between sets).

3.3 XML and semistructured data

XML has become an increasingly popular data-format embraced by a lot of different communities. XML is extremely attractive because it offers a simple, intuitive and uniform text-based syntax and is extensible. One can find today XML proposals for messages, text content delivery and presentation, data content, documents, software components, scientific data, real-estate ads, financial products, cooking recipes, etc. Unfortunately this also means that XML is far too general and if people plan to use it in serious applications (mainly for Electronic Document Interchange, in a broad sense), they will need to provide a specification (i.e. structure, constraints, etc.) for their XML, which XML itself cannot offer. In order to specify and enforce this structure, people have been using Document Type Definitions (DTDs), inherited from SGML and more recently, XML Schema.

Buneman, Davidson, Fan, Hara, and Tan. have investigated integrity constraints for XML data. Both DTDs and the XML Schema porposal lack a clean and general treatment of key dependencies. We discuss the definition of keys for XML documents, paying particular attention to the concept of a relative key, which is commonly used in hierarchically structured documents and scientific databases. We also investigate the (finite) implication problems associated with these dependencies. In contrast to other proposals of keys for XML, these two classes of keys can be reasoned about efficiently. In particular, we show that their (finite) implication problems are finitely axiomatizable and are decidable in polynomial time.

Buneman, Deutsch and Tan have worked on a deterministic model for semistructured data and Buneman and Pierce have worked on union types for semistructured data. Semistructured databases are treated as dynamically typed: they come equipped with no independent schema or type system to constrain the data. Query languages that are designed for semistructured data, even when used

with structured data, typically ignore any type information that may be present. The consequences of this are what one would expect from using a dynamic type system with complex data: fewer guarantees on the correctness of applications. For example, a query that would cause a type error in a statically typed query language will silently return the empty set when applied to a semistructured representation of the same data. We describe a system of untagged UNION TYPES that can accommodate variations in structure while still allowing a degree of static type checking.

Sahuguet has obtained some preliminary results that explore how DTDs are being used for specifying the structure of XML documents. By looking at some publicly available DTDs, we look at how people are actually (mis)using DTDs, show some shortcomings, list some requirements and discuss possible replacements.

Liefke has worked on horizontal query optimization on ordered semistructured data. The exchange and storage of XML data is becoming increasingly important. In contrast to conventional semistructured data, the labels in a document-oriented representation such as XML are ordered. Furthermore, regular expressions (DTDs) describe the horizontal (and vertical) structure. Conventional query languages for semi-structured data ignore the horizontal order and are therefore limited in their expressiveness and optimizability. We describe a query language for querying ordered semistructured data. This query language provides primitives for specifying more powerful queries on ordered semistructured data. Furthermore, we describe how horizontal type information in DTDs is used to optimize queries based on finite automata.

Liefke and Davidson have investigated view maintenance for hierarchical semistructured data. While several important aspects of XML have been investigated, such as query languages, type systems, and storage models, the issue of incrementally maintaining XML views is largely unstudied. XML views differ from relational views in two essential ways: 1) There is no rigid type system, and 2) The query definition often performs complex restructuring far beyond the typical select-project-join query definition in relational views. We address the problem of incrementally maintaining views over XML data with key constraints. We describe a system called WHAX (Warehouse Architecture for XML) that allows the definition and incremental maintenance of views over existing relational and XML data sources with keys. Our query language supports important operations, such as joins, aggregations, regrouping, and restructuring operations such as flattening. We generalize several well-known results about view maintenance in the relational model based on the notion of "multi-linearity". Furthermore, we demonstrate how incremental view maintenance improves the efficiency for XML views defined on real XML data.

3.4 Data Provenance and Annotation

Buneman and Tan (with Khanna) have investigated definitions and properties of the data provenance concept. With the proliferation of database views and curated databases, the issue of *data provenance* – where a piece of data came from and the process by which it arrived in the database – is becoming increasingly important, especially in scientific databases where understanding provenance is crucial to the accuracy and currency of data. We describe an approach to computing provenance when the data of interest has been created by a database query. We adopt a syntactic approach and present results for a general data model that applies to relational databases as well as to hierarchical data such as XML. A novel aspect of our work is a distinction between "why" provenance (refers to the source data that had some influence on the existence of the data) and "where" provenance (refers to the location(s) in the source databases from which the data was extracted).

Buneman and Tan (with Bird) have investigated the design of a query language for annotation graphs. The multidimensional, heterogeneous, and temporal nature of speech databases raises interesting challenges for representation and query. Recently, annotation graphs have been proposed as a general-purpose representational framework for speech databases. Typical queries on annotation graphs require path expressions similar to those used in semistructured query languages. However, the underlying model is rather different from the customary graph models for semistructured data: the graph is acyclic and unrooted, and both temporal and inclusion relationships are important. We develop a query language and describe optimization techniques for an underlying relational representation.

3.5 Updates

Davidson and Liefke have worked on the problem of maintaining derived data in the context of database changes. "View maintenance" describes the problem of maintaining a materialized view while updating the source database(s). Updates to the source database are either immediately propagated to the view or are accumulated over time and the view is updates in frequent intervals (for instance, during night). "View update" is the problem of propagating updates to the view to the source database.

They have developed a generic update language, CPL+, for updating complex value databases – databases containing values composed of base values, sets, tuples, and variants. The complex value model is a generalization of the relational model. We propose various simplification and optimizations so that an update on a given database is transformed into a more efficient update expression. Further, they extended this work to the object-oriented data model. A new language, OQL+, has been developed to specify updates for such databases in the flavor of OQL and the update primitives known from SQL. Interesing issues such as efficient execution, non-determinstism of updates, and cost-based optimizations are investigated in this project.

3.6 Integrated Access to Genomic Data Sources

Davidson, Tannen, et al, have performed and reported on experiments in applications of databases to bioinformatics. The integration of heterogeneous data sources and software systems is a major issue in the biomedical community and several approaches have been explored: linking databases, "on-the-fly" integration through views, and integration through warehousing. We report on our experiences with two systems that were developed at the University of Pennsylvania: an integration system called K2, which has primarily been used to provide views over multiple external data sources and software systems; and a data warehouse called GUS which downloads, cleans, integrates and annotates data from multiple external data sources. Although the view and warehouse approaches each have their advantages, there is no clear "winner". Therefore, users must consider how the data is to be used, what the performance guarantees must be, and how much programmer time and expertise is available to choose the best strategy for a particular application. Our experiences also point to some practical tips on how updates should be published by the community, and how XML can be used to facilitate the processing of updates in a warehousing environment.

Davidson and Liefke (with Limsoon Wong) have investigated creating and maintaining curated view databases. The process of building a new database relevant to some field of study in biology involves transforming, integrating, and cleansing multiple external data sources, as well as adding new material and annotations. Creating and maintaining these "view" databases raise a number

of problems: 1) How can we specify and implement the transformation and integration from the underlying source databases to the view database? 2) How can we automate the refresh process? 3) How can we track the origins or "provenance" of data? The work discusses these phases of creating and maintaining curated view databases and contrast solutions where appropriate.

References

[1] A. Kosky and Susan Davidson and Peter Buneman. Semantics of Database Transformations. In L. Libkin and B. Thalheim, editor, *Semantics of Databases*. Springer LNCS 1358, Feb 1998.

[2] Alin Deutsch and Lucian Popa and Val Tannen. Physical Data Independence, Constraints and Optimization with Universal Plans. In *International Conference on Very Large Databases (VLDB)*, 1999.

[3] Alin Deutsch and Lucian Popa and Val Tannen. Chase & Backchase: A Method for Query Optimization with Materialized Views and Integrity Constraints. Technical Report MS-CIS-01-16, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104, 2001.

[4] Alin Deutsch and Mary Fernandez and Dan Suciu. Storing semistructured data with STORED. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, June 1999.

[5] Alin Deutsch and Mary Fernandez and Daniela Florescu and Alon Levy and Dan Suciu. XML-QL: A Query Language for XML. In *W3C Note*, August 1998.

[6] Alin Deutsch and Val Tannen. Containment and Integrity Constraints for XPath Fragments. In *KRDB 2001*, 2001.

[7] Alin Deutsch and Val Tannen. Optimization Properties for Classes of Conjunctive Regular Path Queries. In *DBPL'01*, 2001.

[8] Arnaud Sahuguet. Everything You Ever Wanted to Know About DTDs, But Were Afraid to Ask. In *WebDB-2000*, 2000.

[9] Arnaud Sahuguet and Benjamin Pierce and Val Tannen. Distributed Query Optimization: Can Mobile Agents Help? Technical Report MS-CIS-00-??, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, Pa 19104, 2000.

[10] Arnaud Sahuguet and Fabien Azavant. Building light-weight wrappers for legacy Web data-sources using W4F. In *International Conference on Very Large Databases (VLDB)*, 1999.

[11] Arnaud Sahuguet and Fabien Azavant. Looking at the Web through XML glasses. In *CoopIs'99*, 1999.

[12] Arnaud Sahuguet and Fabien Azavant. Web Ecology: Recycling HTML pages as XML documents using W4F. In *WebDB'99*, 1999.

[13] Arnaud Sahuguet and Val Tannen. ubQL, a Language for Programming Distributed Query Systems. In *webDB*, 2001.

[14] Arnaud Sahuguet and Val Tannen. Resource Sharing Through Query Process Migration. Technical Report MS-CIS-01-10, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104, 2001.

[15] Peter Buneman, Wenfei Fan, Jérôme Siméon, and Scott Weinstein. Constraints for Semistructured Data and XML. *SIGMOD Record*, 30(1), March 2001.

[16] Peter Buneman, Wenfei Fan, and Scott Weinstein. Path Constraints in Semistructured Databases. *Journal of Computer and System Sciences*, 61(2):146–193, 2000.

[17] Carmem Hara and Susan Davidson. Inference Rules for Nested Functional Dependencies. Technical Report MS-CIS-98-19, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, Pa 19104, 1998.

[18] Carmem Hara and Susan Davidson. Reasoning about Nested Functional Dependencies. In *Proceedings of ACM Symposium on Principles of Database Systems (PODS)*, Jun 1999.

[19] S. Davidson, J. Crabtree, B. Brunk, J. Schug, V. Tannen, C. Overton, and C. Stoeckert. K2/Kleisli and GUS: Experiments in Integrated Access to Genomic Data Sources. *IBM SYstems Journal*, 2001.

[20] Hartmut Liefke and Dan Suciu. XMill: an Efficient Compressor for XML Data. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, Jun 2000.

[21] Hartmut Liefke and Susan Davidson. An Execution Model for CPL+. Technical Report MS-CIS-98-29, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, Pa 19104, 1998.

[22] Hartmut Liefke and Susan Davidson. Updating Complex Value Databases. Technical Report MS-CIS-98-06, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, Pa 19104, 1998.

[23] Hartmut Liefke and Susan Davidson. Processing Updates on Complex Value Databases. In *Information Resource Management Association International Conference*, May 1999.

[24] Hartmut Liefke and Susan Davidson. Specifying Updates in Biomedical Databases. In *SSDBM'99*, 1999.

[25] Lucian Popa. Object-Relational Query Optimization with Chase and Backchase, 2000. PhD Dissertation, University of Pennsylvania.

[26] Lucian Popa and Alin Deutsch and Arnaud Sahuguet and Val Tannen. A Chase Too Far? In *Proceedings of ACM SIGMOD International Conference on Management of Data*, Dallas, USA, May 2000.

[27] Lucian Popa and Val Tannen. Chase and axioms for PC queries and dependencies. Technical Report MS-CIS-98-34, University of Pennsylvania, 1998.

[28] Lucian Popa and Val Tannen. An Equational Chase for Path-Conjunctive Queries, Constraints, and Views. In *International Conference on Database Theory (ICDT)*, Jerusalem, Israel, January 1999.

[29] Mary Fernandez and WangChiew Tan and Dan Suciu. SilkRoute: Trading between Relations and XML. In *WWW9*, May 2000.

[30] Peter Buneman and Alin Deutsch and Wang-Chiew Tan. A deterministic model for semistructured data. In *Workshop on Query Processing for Semistructured Data and Non-Standard Data Formats*, 1998.

[31] Peter Buneman and Alin Deutsch and WangChiew Tan. A deterministic model for semistructured data. In *Workshop on Query Processing for Semistructured Data and Non-Standard Data Formats*, 1999.

[32] Peter Buneman and Alin Deutsch and Wenfei Fan and Hartmut Liefke and Arnaud Sahuguet and Wang-Chiew Tan. Beyond XML Query Languages. In *Query Language Workshop (QL'98)*, Nov 1998.

[33] Peter Buneman and Benjamin Pierce. Union Types for Semistructured Data. In *DBPL*, 1999.

[34] Peter Buneman and Jonathan Crabtree and Susan Davidson and Val Tannen and L. Wong. BioKleisli. In S. Letovsky, editor, *Bioinformatics*. Kluwer Academic Publishers, 1998.

[35] Peter Buneman and Mary Fernandez and Dan Suciu. UnQL: A Query Language and Algebra for Semistructured Data Based on Structural Recursion. *VLDB Journal*, 9(1):76–110, 2000.

[36] Peter Buneman and Sanjeev Khanna and Wang-Chiew Tan. Data Provenance: Some Basic Issues. In *Foundations of Software Technology and Theoretical Computer Science*, 2000.

[37] Peter Buneman and Sanjeev Khanna and Wang-Chiew Tan. Why and Where: A Characterization of Data Provenance. In *International Conference on Database Theory (ICDT)*, 2001.

[38] Peter Buneman and Susan Davidson and Wenfei Fan and Carmem Hara and Wang-Chiew Tan. Keys for XML. In *WWW10*, May 2001.

[39] Peter Buneman and Susan Davidson and Wenfei Fan and Carmem Hara and Wang-Chiew Tan. Reasoning about Keys for XML (Technical Report). In *International Workshop on Database Programming Languages (DBPL)*, 2001.

[40] Peter Buneman and Wenfei Fan and Scott Weinstein. Equality, Type and Word Constraints. Technical Report MS-CIS-98-32, University of Pennsylvania, 1998.

[41] Peter Buneman and Wenfei Fan and Scott Weinstein. Interaction between Path and Type Constraints. Technical Report MS-CIS-98-16, University of Pennsylvania, 1998.

[42] Peter Buneman and Wenfei Fan and Scott Weinstein. Path Constraints on Deterministic Graphs. Technical Report MS-CIS-98-33, University of Pennsylvania, 1998.

[43] Peter Buneman and Wenfei Fan and Scott Weinstein. Path Constraints on Semistructured and Structured Data. In *PODS*, Jun 1998.

[44] Peter Buneman and Wenfei Fan and Scott Weinstein. Interaction between Path and Type Constraints. In *Proceedings of ACM Symposium on Principles of Database Systems (PODS)*, Jun 1999.

[45] Peter Buneman and Wenfei Fan and Scott Weinstein. Query Optimization for Semistructured Data using Path Constraints in a Deterministic Data Model. In *DBPL*, 1999.

[46] S. Davidson and H. Liefke. View Maintenance for Hierarchical Semistructured Data. In *Proceedings of DaWak'00*, London, England, September 2000.

[47] S. Davidson and H. Liefke and L. Wong. Creating and Maintaining Curated View Databases. In *Knowledge Discovery in Biology Databases.* World Scientific Publishing Company, 2000.

[48] Steven Bird and Peter Buneman and Wang-Chiew Tan. Towards a query language for annotation graphs. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, 2000.

[49] Wenfei Fan. Path Constraints for Databases with or without Schemas, 1999. PhD Dissertation.

[50] Zo Lacroix and Arnaud Sahuguet and Raman Chandrasekar. Information Extraction and Database Techniques: A User-Oriented Approach to Querying the Web. In *CAiSE 1998*, 1998.

DEPARTMENT OF THE ARMY
U.S. ARMY ROBERT MORRIS ACQUISITION CENTER
RESEARCH TRIANGLE PARK CONTRACTING DIVISION
P.O. BOX 12211
RESEARCH TRIANGLE PARK, NC 27709-2211

REPLY TO
ATTENTION OF:

April 18, 2002

AMSSB-ACR P - 38743-CI

Subject: Research Agreement No. DAAG55-98-1-0331

Dept. of Computer & Information Sciences
University of Pennsylvania
200 South 33rd St.
Philadelphia, PA 19104-6389

Dear Professor Val Tannen:

PAST DUE NOTICE

The Army Research Office (ARO) has not received a final technical report for Research Agreement No. DAAG55-98-1-0331 which was due on August 31, 2001. Failure to comply with the procedures as outlined in the agreement could impact future funding. Please submit this report without delay, to the US Army Research Office, ATTN: AMSRL-RO-DS (Technical Reports), P.O. Box 12211, Research Triangle Park, NC 27709-2211. When using a commercial carrier for overnight delivery, please submit the report to the US Army Research Office, ATTN: AMSRL-RO-DS (Technical Reports), 4300 S. Miami Blvd., Durham, NC 27703-9142. Electronic files in Portable Document Format (PDF) will be accepted when submitted via email to reports@aro.arl.army.mil.

If your research agreement references the ARO Form 18 Reporting Instructions, it is available from the ARO's web site. The web site address is: http://www.aro.army.mil. The Form SF 298, Report Documentation Page, and the Memorandum of Transmittal can also be downloaded.

If you have questions or believe you have erroneously received this notification, please notify Mary Jackson via email to jackson@aro.arl.army.mil.

Sincerely,

/s/
Larry E. Travis
Contracting Officer

Copy Furnished Technical Monitor:
David Hislop

DEPARTMENT OF THE ARMY
ARMY RESEARCH OFFICE
P.O. BOX 12211
RESEARCH TRIANGLE PARK, NC 27709-2211

REPLY TO
ATTENTION OF

December 6, 2001

AMSRL-RO-BI

SUBJECT: 38743-CI

Professor Val Tannen
Dept. of Computer & Information Sciences
University of Pennsylvania
200 South 33rd St.
Philadelphia, PA USA 19104-6389

Dear Professor Tannen :

1. As the designated Principal Investigator of agreement number DAAG559810331, one of your prime and vital responsibilities is full compliance with the Army Research Office's reporting requirements. These requirements, as referenced in your agreement, are described in the ARO Form 18, "REPORTING INSTRUCTIONS". Your technical reports provide valuable information to the ARO Project Monitor, and are the deliverables which justify payments under your agreement.

2. The ARO Form 18 is available on ARO Web Site under the following URL: http://www.aro.army.mil. Once connected to the ARO Home Page, select "Forms", then click on "ARO Form 18 Reporting Instructions." From there, it is self-explanatory. All future updates of instructions will be located on the ARO Web Site and should be checked periodically for new information.

3. If you are unable to download a copy of the ARO Form 18, please send an E-mail message to ipr@arl.aro.army.mil and request a hard copy mailed to you. If the address on this letter is incomplete or inaccurate, please include your full mailing address.

/s/
Bessie B. Oakley
Chief, Information Control and Analysis Office